

**Notice of Allowability**

Application No.

10/724,921

Applicant(s)

KAWAGUCHI ET AL.

Examiner

Anil Khatri

Art Unit

2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to 10/2/2007.
2. ☒ The allowed claim(s) is/are 1, 3-20, 22-28, 30-45 and 47-61.
3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
  - a) ☐ All   b) ☐ Some\*   c) ☐ None   of the:
    1. ☐ Certified copies of the priority documents have been received.
    2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
    3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

\* Certified copies not received: \_\_\_\_\_.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.  
**THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.**

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
  5. ☐ CORRECTED DRAWINGS ( as "replacement sheets") must be submitted.
    - (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review ( PTO-948) attached
      - 1) ☐ hereto or 2) ☐ to Paper No./Mail Date \_\_\_\_\_.
    - (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date \_\_\_\_\_.
- Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

**Attachment(s)**

1. ☒ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO/SB/08),  
Paper No./Mail Date \_\_\_\_\_
4. ☐ Examiner's Comment Regarding Requirement for Deposit  
of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☐ Interview Summary (PTO-413),  
Paper No./Mail Date \_\_\_\_\_
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other \_\_\_\_\_

### **EXAMINER'S AMENDMENT**

An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it **MUST** be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Robert Kowert, Registration no. 39,255 on 11/13/2007.

The application has been amended as follows:

#### **In Claims**

1. (Currently amended) A system, comprising:

one or more processors; and

a memory storing program instructions, wherein the program instructions are executable by the one or more processors to implement:

a virtual machine;

a plurality of subsystems configured to execute within the virtual machine, wherein two or more of the plurality of subsystems provide different versions of an isomorphic interface to functions of the subsystems; and

a proxy mechanism configured to generate, for one of the plurality of subsystems at runtime of the one of the plurality of subsystems, a proxy to a version of the isomorphic interface provided by a particular one of the two or more subsystems;

wherein the proxy is configured to:

receive a call to the isomorphic interface from the one of the plurality of subsystems;

convert the call in accordance with the version of the isomorphic interface provided by the particular one of the two or more subsystems; and

forward the converted call to the particular one of the two or more subsystems for execution; and

wherein the proxy mechanism is further configured to generate, for another one of the plurality of subsystems at runtime of the one of the plurality of subsystems, another proxy to a different version of the isomorphic interface provided by another one of the two or more subsystems.

2. (Canceled)

3. (Previously presented) The system as recited in claim 1, wherein, to generate a proxy to a version of the isomorphic interface provided by a particular one of the two or more subsystems, the proxy mechanism is further configured to determine that the particular one of the two or more subsystems provides a correct version of the isomorphic interface for the one of the plurality of subsystems.

4. (Previously presented) The system as recited in claim 3, wherein the one of the

Art Unit: 2191

plurality of subsystems is configured to specify the isomorphic interface to be proxied to the proxy mechanism, and wherein the proxy mechanism is further configured to determine that the particular one of the two or more subsystems provides the correct version of the isomorphic interface according to said specification and to generate the proxy according to said specification.

5. (Previously presented) The system as recited in claim 1, wherein the proxy mechanism is further configured to generate another proxy configured to return results of said execution from the particular one of the two or more subsystems to the one of the plurality of subsystems.

6. (Previously presented) The system as recited in claim 1, wherein the proxy mechanism is further configured to:

receive the call to the isomorphic interface from the one of the plurality of subsystems;  
and

generate the proxy to the version of the isomorphic interface provided by the particular one of the two or more subsystems in response to said call to the isomorphic interface.

7. (Previously presented) The system as recited in claim 1, wherein the proxy mechanism is further configured to provide an interface to the proxy mechanism for the plurality of subsystems, wherein the interface is readable by the subsystems to specify isomorphic interfaces provided by other ones of the subsystems to be proxied to by the proxy mechanism.

8. (Previously presented) The system as recited in claim 1, wherein the proxy is further configured to convert the call in accordance with the version of the isomorphic interface provided by the particular one of the two or more subsystems using Java Reflection.

Art Unit: 2191

9. (Original) The system as recited in claim 1, wherein the virtual machine is a Java Virtual Machine (JVM).

10. (Original) The system as recited in claim 1, wherein the one of the plurality of subsystems is an application, and wherein the two or more subsystems are versions of a runtime library.

11. (Original) The system as recited in claim 1, wherein the one of the plurality of subsystems and the two or more subsystems are applications.

12. (Original) The system as recited in claim 1, wherein the plurality of subsystems are mobile agents.

13. (Previously presented) A system, comprising:

one or more processors; and

a memory storing program instructions, wherein the program instructions are executable by the one or more processors to implement:

a virtual machine;

a plurality of subsystems configured to execute within the virtual machine; and

a proxy mechanism configured to:

generate a proxy for a version of an interface between two of the plurality of subsystems at runtime of at least one of the two subsystems, wherein the proxy is configured to convert calls between the two subsystems in

accordance with the version of the interface; and

generate another proxy for another version of the interface between another two of the plurality of subsystems at runtime of at least one of the other two of the plurality of subsystems, wherein the other proxy is configured to convert calls between the other two of the plurality of subsystems in accordance with the other version of the interface.

14. (Original) The system as recited in claim 13, wherein the proxy is configured to:

receive from a first of the two subsystems a call to a second of the two subsystems;

convert the call in accordance with the version of the interface; and

forward the converted call to the second subsystem for execution by the second subsystem.

15. (Original) The system as recited in claim 14, wherein the proxy mechanism is further configured to generate another proxy configured to return results of said execution from the second subsystem to the first subsystem.

16. (Original) The system as recited in claim 14, wherein the proxy mechanism is further configured to generate the proxy for the version of the interface between the two subsystems in response to said call to the second subsystem.

17. (Original) The system as recited in claim 14, wherein the proxies are configured to convert the calls between the subsystems using Java Reflection.

18. (Original) The system as recited in claim 13, wherein a first of the two subsystems is configured to specify the interface to be proxied to the proxy mechanism, and wherein the proxy

Art Unit: 2191

mechanism is further configured to generate the proxy for the interface between the two subsystems in accordance with said specification.

19. (Original) The system as recited in claim 13, wherein the virtual machine is a Java Virtual Machine (JVM).

20. (Currently amended) A system, comprising:

a processor; and

a memory comprising program instructions, wherein the program instructions are executable by the processor to implement a proxy mechanism configured to generate, for a subsystem at runtime of the subsystem, a proxy to a version of an isomorphic interface provided by a particular one of a plurality of subsystems that provide versions of the isomorphic interface;

wherein the proxy is configured to:

receive a call to the isomorphic interface from the subsystem;

convert the call in accordance with the version of the isomorphic interface provided by the particular one of the plurality of subsystems; and

forward the converted call to the particular one of the plurality of subsystems for execution; and

wherein the proxy mechanism is further configured to generate, for a different subsystem at runtime of a different subsystem, another proxy to another one of the plurality of subsystems that provides a different version of the isomorphic interface.

21. (Canceled)

22. (Previously presented) The system as recited in claim 20, wherein the proxy mechanism is further configured to generate another proxy configured to return results of said execution from the particular one of the plurality of subsystems to the subsystem.

23. (Previously presented) The system as recited in claim 20, wherein the subsystem is configured to specify the isomorphic interface to be proxied to the proxy mechanism, and wherein the proxy mechanism is further configured to generate the proxy according to said specification.

24. (Original) The system as recited in claim 20, wherein the program instructions are executable by the processor to implement a virtual machine in the memory, wherein the proxy mechanism, the subsystems, and the proxy are configured to operate within the virtual machine.

25. (Previously presented) The system as recited in claim 24, wherein the virtual machine is a Java Virtual Machine (JVM).

26. (Previously presented) The system as recited in claim 20, wherein the proxy is further configured to convert the call in accordance with the version of the isomorphic interface provided by the particular one of the plurality of subsystems using Java Reflection.

27. (Currently amended) A computer system, comprising:

means for generating a proxy to a version of an isomorphic interface provided by two or more subsystems for one or more other subsystems in a virtual machine at runtime of at least one of the one or more other subsystems;



means for generating another proxy to another version of the isomorphic interface provided by the two or more subsystems for the one or more other subsystems in the virtual machine at runtime of at least one of the one or more other subsystems;

means for the proxies to convert calls from the one or more other subsystems to the versions of the isomorphic interface provided by the two or more subsystems; and

means for the proxies to forward the converted calls to the two or more subsystems for execution.

28. (Currently amended) A computer-implemented method, comprising:

a proxy mechanism generating, for a subsystem at runtime of the subsystem, a proxy to a version of an isomorphic interface provided by a particular one of a plurality of subsystems that provide versions of the isomorphic interface;

the proxy receiving a call to the isomorphic interface from the subsystem;

the proxy converting the call in accordance with the version of the isomorphic interface provided by the particular one of the plurality of subsystems;

the proxy forwarding the converted call to the particular one of the plurality of subsystems for execution; and

the proxy mechanism generating another proxy to another one of the plurality of subsystems that provides a different version of the isomorphic interface for another subsystem at runtime of the other subsystem.

29. (Canceled)

30. (Previously presented) The computer-implemented method as recited in claim 28, further comprising:

the proxy mechanism generating another proxy configured to return results of executing the call from the particular one of the plurality of subsystems to the subsystem;

the particular one of the plurality of subsystems executing the call; and

the other proxy returning results of said executing the call to the subsystem.

31. (Previously presented) The computer-implemented method as recited in claim 28, further comprising the subsystem specifying the isomorphic interface to be proxied to the proxy mechanism, wherein the proxy mechanism generates the proxy to the particular one of the plurality of subsystems according to said specification.

32. (Previously presented) The computer-implemented method as recited in claim 28, further comprising the proxy mechanism receiving the call to the isomorphic interface, wherein the proxy mechanism generates the proxy to the particular one of the plurality of subsystems in response to said receiving the call.

33. (Previously presented) The computer-implemented method as recited in claim 28, wherein the proxy converts the call using Java Reflection.

34. (Previously presented) The computer-implemented method as recited in claim 28, wherein the proxy mechanism, the subsystems, and the proxy are implemented within a computer-implemented virtual machine.

35. (Previously presented) The computer-implemented method as recited in claim 28, wherein the subsystem is an application, and wherein the plurality of subsystems are versions of

Art Unit: 2191

a runtime library.

36. (Previously presented) The computer-implemented method as recited in claim 28, wherein the subsystem and the plurality of subsystems are applications.

37. (Previously presented) The computer-implemented method as recited in claim 28, wherein the subsystem and the plurality of subsystems are mobile agents.

38. (Previously presented) A computer-implemented method, comprising:

generating a proxy for a version of an interface between two of a plurality of subsystems in a virtual machine at runtime of at least one of the two subsystems, wherein the proxy is configured to convert calls between the two subsystems in accordance with the version of the interface; and

generating another proxy for another version of the interface between another two of the plurality of subsystems at runtime of at least one of the other two of the plurality of subsystems, wherein the other proxy is configured to convert calls between the other two of the plurality of subsystems in accordance with the other version of the interface.

39. (Previously presented) The computer-implemented method as recited in claim 38, further comprising:

the proxy receiving from a first of the two subsystems a call to a second of the two subsystems;

the proxy converting the call in accordance with the version of the interface; and

Art Unit: 2191

the proxy forwarding the converted call to the second subsystem for execution by the second subsystem.

40. (Previously presented) The computer-implemented method as recited in claim 39, wherein the proxy converts the call in accordance with the version of the interface using Java Reflection.

41. (Previously presented) The computer-implemented method as recited in claim 39, further comprising generating another proxy configured to return results of said execution from the second subsystem to the first subsystem.

42. (Previously presented) The computer-implemented method as recited in claim 39, further comprising generating the proxy for the version of the interface between the two subsystems in response to said call to the second subsystem.

43. (Previously presented) The computer-implemented method as recited in claim 38, further comprising:

a first of the two subsystems specifying the interface to be proxied; and

generating the proxy for the interface between the two subsystems in accordance with said specification.

44. (Previously presented) The computer-implemented method as recited in claim 38, wherein the virtual machine is a Java Virtual Machine (JVM).

45. (Currently amended) A computer-readable storage media comprising program instructions, wherein the program instructions are computer-executable to implement:

a proxy mechanism generating, for a subsystem at runtime of the subsystem, a proxy to a version of an isomorphic interface provided by a particular one of a plurality of subsystems that provide versions of the isomorphic interface;

the proxy receiving a call to the isomorphic interface from the subsystem;

the proxy converting the call in accordance with the version of the isomorphic interface provided by the particular one of the plurality of subsystems;

the proxy forwarding the converted call to the particular one of the plurality of subsystems for execution; and

the proxy mechanism generating another proxy to another one of the plurality of subsystems that provides a different version of the isomorphic interface for another subsystem at runtime of the other subsystem.

46. (Canceled)

47. (Previously presented) The computer-readable storage media as recited in claim 45, wherein the program instructions are further computer-executable to implement:

the proxy mechanism generating another proxy configured to return results of executing the call from the particular one of the plurality of subsystems to the subsystem;

the particular one of the plurality of subsystems executing the call; and

the other proxy returning results of said executing the call to the subsystem.

48. (Previously presented) The computer-readable storage media as recited in claim 45, wherein the program instructions are further computer-executable to implement the

Art Unit: 2191

subsystem specifying the isomorphic interface to be proxied to the proxy mechanism, wherein the proxy mechanism generates the proxy to the particular one of the plurality of subsystems according to said specification.

49. (Previously presented) The computer-readable storage media as recited in claim 45, wherein the program instructions are further computer-executable to implement the proxy mechanism receiving the call to the isomorphic interface, wherein the proxy mechanism generates the proxy to the particular one of the plurality of subsystems in response to said receiving the call.

50. (Previously presented) The computer-readable storage media as recited in claim 45, wherein the proxy converts the call using Java Reflection.

51. (Previously presented) The computer-readable storage media as recited in claim 45, wherein the proxy mechanism, the subsystems, and the proxy are implemented within a virtual machine.

52. (Previously presented) The computer-readable storage media as recited in claim 45, wherein the subsystem is an application, and wherein the plurality of subsystems are versions of a runtime library.

53. (Previously presented) The computer-readable storage media as recited in claim 45, wherein the subsystem and the plurality of subsystems are applications.

54. (Previously presented) The computer-readable storage media as recited in claim 45, wherein the subsystem and the plurality of subsystems are mobile agents.

55. (Previously presented) A computer-readable storage media comprising program instructions, wherein the program instructions are computer-executable to implement:

generating a proxy for a version of an interface between two of a plurality of subsystems in a virtual machine at runtime of at least one of the two subsystems, wherein the proxy is configured to convert calls between the two subsystems in accordance with the version of the interface; and

generating another proxy for another version of the interface between another two of the plurality of subsystems at runtime of at least one of the other two of the plurality of subsystems, wherein the other proxy is configured to convert calls between the other two of the plurality of subsystems in accordance with the other version of the interface.

56. (Previously presented) The computer-readable storage media as recited in claim 55, wherein the program instructions are further computer-executable to implement:

the proxy receiving from a first of the two subsystems a call to a second of the two subsystems;

the proxy converting the call in accordance with the version of the interface; and

the proxy forwarding the converted call to the second subsystem for execution by the second subsystem.

57. (Previously presented) The computer-readable storage media as recited in claim 56, wherein the proxy converts the call in accordance with the version of the interface using Java Reflection.

58. (Previously presented) The computer-readable storage media as recited in claim 56, wherein the program instructions are further computer-executable to implement generating another proxy configured to return results of said execution from the second subsystem to the

Art Unit: 2191

first subsystem.

59. (Previously presented) The computer-readable storage media as recited in claim 56, wherein the program instructions are further computer-executable to implement generating the proxy for the version of the interface between the two subsystems in response to said call to the second subsystem.

60. (Previously presented) The computer-readable storage media as recited in claim 55, wherein the program instructions are further computer-executable to implement:

a first of the two subsystems specifying the interface to be proxied; and

generating the proxy for the interface between the two subsystems in accordance with said specification.

61. (Previously presented) The computer-readable storage media as recited in claim 55, wherein the virtual machine is a Java Virtual Machine (JVM).

***Reason For Allowance***

Claims 1, 3-20, 22-28, 30-45 and 47-61 (renumbered 1-57) are allowed

The following is an examiner's statement of reasons for allowance:

The cited prior taken alone or in combination does not teach or fairly suggest the claimed invention of proxying isomorphic interfaces in different subsystems. Specifically, claims 1, 13, 20, 27, 28, 38, 45 and 55 are directed to a proxy mechanism configured to generate, for one of the plurality of subsystems at runtime of the one of the plurality of subsystems, a proxy to a version of the isomorphic interface provided by a particular one of the two or more subsystems wherein the proxy is configured to receive a call to the isomorphic interface from the one of the



Art Unit: 2191

plurality of subsystems, convert the call in accordance with the version of the isomorphic interface provided by the particular one of the two or more subsystems, forward the converted call to the particular one of the two or more subsystems for execution and generating a proxy for a version of an interface between two of the plurality of subsystem. Therefore, the claimed invention improves on subsystems which are evolving rapidly, having a common library including preserving backward compatibility between versions of the subsystems, since any change in the shared library may affect systems currently deployed so tedious execution process and errors could be avoided.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

### ***Conclusion***


Any inquiry concerning this communication or earlier communications from the examiner should be directed to Anil Khatri whose telephone number is 571-272-3725. The examiner can normally be reached on M-F 8:30-5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2191

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

\*\*\*

  
**ANIL KHATRI**  
**PRIMARY EXAMINER**